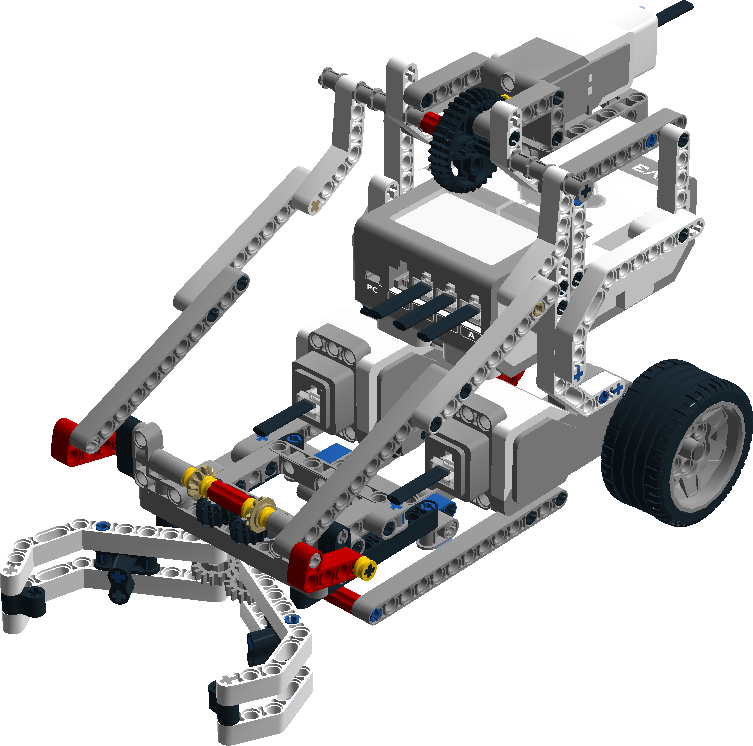
Week 3 - Making a connection



Date: 02-10-2018  
Version 1.3

Contents

[Introduction 3](#_Toc530337298)

[Overview assignment(s) week 3 3](#_Toc530337299)

[Getting the example project to work 4](#_Toc530337300)

[Intermezzo: The DLL 5](#_Toc530337301)

[The Windows forms application 6](#_Toc530337302)

[Design the Graphical user interface- Assignment 6](#_Toc530337303)

[Appendix A: The example applications 9](#_Toc530337304)

[Setting everything up 9](#_Toc530337305)

[Working with the examples 9](#_Toc530337306)

# Introduction

In Fundamentals for ICT Students (block) 2 you’re going to work on a project that will cover all 4 streams! This week we’re going to have a look at ICT & Software and ICT & Media design and how they fit in realizing an IT solution.

As a software engineer you’re going to be involved in anything that is related to software development. You start specialising when you get a job. Working at a company you’re often asked to continue with an existing project or make use of existing components.

On the other hand, as Media designer you will be using different methods to get the right information for designing new media or improving the current design for a better user experience.

To this end this week’s practical is going to be about including existing software (a DLL) in a Windows Forms application and designing a Graphical user interface that is easy to use and intuitive.

The structure of this week’s practical will be as following:

* Create a Windows Forms Application;
* Create a Lego Mindstorms program;
* Design a Windows Forms Application.

## Overview assignment(s) week 3

**Deliverable:** A beautiful Windows forms application working as ‘remote‘ with your Lego Mindstorms robot part.  
**Deliverable per:** 4 Students.  
**Deadline:** Before week 4; you will show it to the teacher during class.

This week you’re going to make a Windows Form Application that can send messages to the brick and read message from the brick. To be able to do this you need to have a computer with Bluetooth. If you do not have access to such a computer, you can borrow a USB Bluetooth dongle from the IISD.

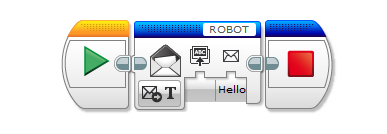


Figure 1: Lego Mindstorms - Messaging

# Getting the example project to work

On Sharepoint you can find a [file](https://portal.fhict.nl/es/FIS2/Week%203/Resources/EV3MessengerWithExampleProjects.rar) which can be extracted. To get access to two example projects. These two projects combine will showcase how you can have a Windows Form Application connecting and ‘communicating’ with the Lego brick.

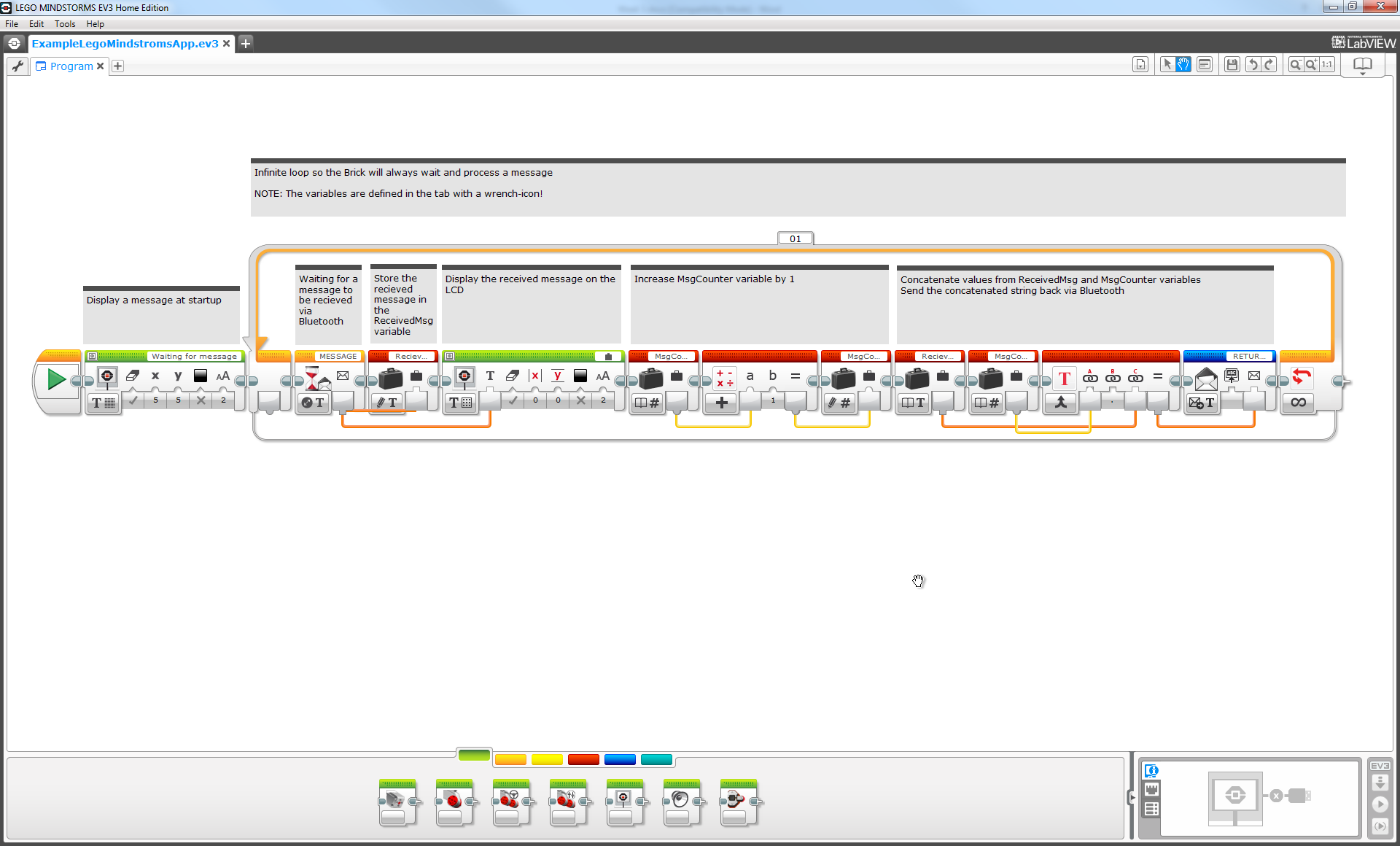
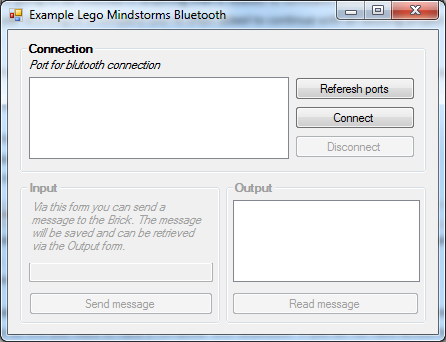


Figure 2: Example Windows Form Application & Lego Mindstroms application

The Windows Form Application uses an external DLL to set up and communicate with the brick via Bluetooth. This week will be about you creating your own Windows and Lego application.

To get started it is very important to get the two example projects working on your laptop and brick, before continue with the practical! In *Appendix A: The example applications* you can find instructions and pointers how to do this.

# Intermezzo: The DLL

To be able to communicate with the brick you’re going to make use of a supplied piece of software (DLL). This software enables you to connect with the brick via Bluetooth and communicate via messages.

The file you extracted also contains two files with the names ‘EV3MessengerLib.dll’ & ‘EV3MessengerLib.XML’. The file ending with .dll is where the compiled (CIL) code is located, whereas the .xml file has the ‘help text/instructions’ to help you with what each method does.

Only the DLL file is really required (no DLL == not working application), but for development purposes we advise you to also include the .xml file into your project.

NOTE: The combination of .dll and .xml is created by Visual Studio when a project is compiled/published.

Keep in mind that the DLL will automatically be included in the executable. This has to do with how the (default way a) reference is created in the project and how the C# compiler works. As indicated in the image below the C# compiler creates the executable by including ‘everything’ of your project.

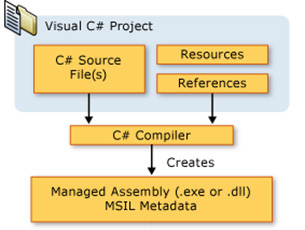


Figure 3: C# Project workflow

# The Windows forms application

Now you know where to find the DLL it’s up to you to create your own Windows Forms application to make use of it. The application should function as a (remote) trigger for your robot to start a ‘process’ on your brick. Depending on which part you worked on last week you need to be able to the de following via your Windows application:

|  |  |  |  |
| --- | --- | --- | --- |
| **Group** | | | |
| Pair 1 (Drive system) | | Pair 2 (Pickup system) | |
| Student 1 | Student 2 | Student 3 | Student 4 |
|  | | C:\Users\888602\Dropbox\FIS\Sharepoint\Block 2\Practical\Robot assignment\Grabber With Frame.png | |
| Requirements:   * Button to:   + Start driving   + Stop driving * A control to see the status :   + Is the robot idle   + Is the robot driving | | Requirements:   * Buttons to:   + Start pickup   + Stop pickup * A control to see the status :   + Is the robot idle   + Is the robot in it’s pick-up process | |

Make sure the Lego program you created last week is receiving and processing the commands. For example: if you press the “Start pickup”-button the pickup system should start the pickup logic you created last week.

You can and should use both the supplied example applications as reference/resource. For more information about these example applications see Appendix A.

## Design the Graphical user interface- Assignment

As the client requested you must make a remote-control application that can control the robot and shows its status, but what functionality should it have? How should it look like? Does client have any design preferences or not? These questions and much more must be answered before you create your final product.

But first let’s get familiar with the application. Add following functionality to the Windows form application:

1. 4 buttons to send commands to the AGV to move Forward, Backward, Left, Right.
2. Pick up the box
3. Drop off the box

In order to satisfy the client with your final product you must first understand what he needs. Then you can design your prove of concept and test it as much as necessary and improve it each time till you have the best possible solution. These are 3 phases of a good design:

1. **User study:** Your users are not you, so you have to research your users.
2. **Design:** it’s all about the design of the application which can gives an impression to the users and in long term make sure they keep using the app if it is designed in a good manner.
3. **User Test:** considering that you are not your user, it is a must to analyse the end user’s experience with the application to be able to find the possible improvements.

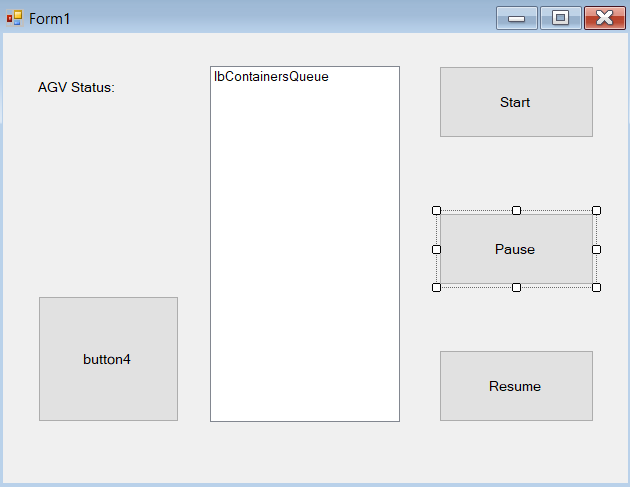
To proceed with each phase, there are several different methods to enhance the design of the final product such as interviewing, online survey, sketching, mock-up, card sorting, prototyping, Guerrilla test, etc.

We are going to take 3 steps:

1. **Get to know your user:** who is going to use the application? What does he want? Why does he want it? How does he want it? You need to answer these questions to be able to design an application that can perform what is expected from it by the user. So fill in the bellow table to get to know your user and its expectations. You can fill in this table using the feature table of week1 assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | What? | Why? | How? |
| 1 | Be able to pick up and drop off containers that weight up to 500kg | In order to transport containers the robot needs to be able to pick them up and also drop them off to do the job correctly | We will use a medium motor to power a claw that grabs and then lifts the container |
| 2 | Move along a predefined path | Because the AGV will be placed in a company which will most likely be busy it is best to have a predefined path that the it follows. So the AGV needs to be able to move along the path with precision to avoid collision and be efficient | We use a colour sensor to detect the colour its currently hovering over and based on that it makes a turn, either right or left. |
| 3 | Be controlled by employee (Start, Resume or Stop the AGV) | Because there isn’t always containers to transport and the person who knows if there are is the employee in charge he should then have control over the machine to insure it does its job correctly | With the use of a C# application we will be able to send messages to the AGV that will then activate the code they refer to |
| 4 | Display AGV status to employee | So that the employee knows if the AGV is free or busy | The C# application will receive a message from the AGV when the status changes |
| 5 | Avoid hitting people, other AGVs and other obstacles | Because a workplace needs to be a place where you feel safe and to feel safe with a big machine nearby you need to be sure that machine wont cause any accidents this would also cost the company money also isn’t good | With the use of touch sensors as bumpers and the ultrasonic sensor to detect nearby objects |
| 6 | Emergency stop | To prevent accidents with the AGV and minimizes the damage of any accidents it does have | When the sensors dectect something that was coded to activate the emergency stop it stops or It can be activated via the C# aplication |
| 7 | Return to start automatically | So that I can transport the next container or be powered off | With code on the AGV when it detects its finished the path it turns around and returns to the start |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |

1. **Sketch the GUI:** Using the information gathered in previous step, draw your first sketch of application’s GUI using pen and paper. Keep it in mind that you are not the user and he may see everything differently. Think about a good way to group by the features based on their characteristics.



1. **Test:** you can test the functionality and see whether you can improve the user experience with an alternative approach? You can use the same feature table and describe how you want to improve it.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | What? | Why? | How? |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |

# Appendix A: The example applications

The examples application form one working solution/are dependent on each other. There is a Visual Studio project in the folder “Visual Studio - example application” and a Lego Mindstorms project in “Lego Mindstorms - example application”.

## Setting everything up



Figure 4: Setup connection between your computer and your Brick

To be able to get the solution working you first need to connect your laptop, via Bluetooth, to the brick (if you do not have Bluetooth you can borrow a USB Bluetooth dongle from the IISD). This Bluetooth connection needs to be done via the configuration screen of your OS. Your brick will ask if you want to connect to the laptop and you need to confirm this and also handle the screen with the password.

TIP: If for some reason your laptop isn’t showing the name of your Brick you can try to disable and enable the Bluetooth on the brick (don’t forget to save in between these two steps).

When the Bluetooth connection between your laptop and brick is created you can start by connecting the USB cable to the brick and start the example Lego Mindstorms application.   
After this you can start the Windows Form application. Be sure you connect via the correct COM port (see the supplied ReadMe.docx).

## Working with the examples

Now everything is set up and you are able to send a text message (via Bluetooth) to the brick and the brick will send a text message (via Bluetooth) back to the Windows Form application as confirmation. See the flow chart below how to use the example applications.



Figure 5: Making use of the example applications